

Esame di Calcolatori Elettronici T

28 Gennaio 2015 (Ing. Informatica)

Esercizio 1

Progettare un sistema basato sul microprocessore DLX dotato di 1 MB di EPROM mappata negli indirizzi bassi, 2 MB di RAM mappata negli indirizzi alti.

Nel sistema è anche presente una periferica a 8 bit, già progettata e denominata INPUT_PORT, in grado di ricevere dati dall'esterno mediante il protocollo di *handshake*. Ogni otto trasferimenti da INPUT_PORT deve essere valutato (mediante una opportuna rete logica) se la somma modulo 256 di tutti i

caratteri letti dalla medesima porta fino a quel momento (i.e., dall'avvio del sistema e incluso il carattere trasferito in quell'istante) è divisibile per 64.

Allorché questo sia verificato, tale la somma deve essere memorizzata in un registro (rete logica) e deve essere contemporaneamente asserita una richiesta di interruzione (rete logica). L'interrupt handler associato a tale evento dovrà,

prima di ogni successivo trasferimento da INPUT_PORT, incrementare via software di una unità un contatore a 32 bit residente in RAM all'indirizzo FFFFFFFF0h, leggere dal registro il valore della somma modulo 256 che risulta divisibile per 64, memorizzare tale valore in FFFFFFFF4h e riattivare l'analisi dei caratteri in ingresso. Nel progetto della rete si consideri il carattere 00000000₂ non divisibile per 64. La procedura appena definita deve essere ripetuta, con la medesima modalità, ogni 8 trasferimenti da INPUT_PORT.

- Indicare i segnali di chip select di tutte le periferiche, memorie e di eventuali ulteriori segnali presenti nel sistema
- Progettare, **ottimizzando le risorse utilizzate**, le reti necessarie al funzionamento del sistema (inclusa la rete che identifica un numero divisibile per 64) indicando inoltre le connessioni di tutte le periferiche e le memorie ai bus del DLX
- Evidenziare, se esistono potenziali criticità nel progetto hardware e indicare eventuali soluzioni che potrebbero essere adottate per eliminarle
- Scrivere il codice dell'*interrupt handler* che gestisce le comunicazioni con la porta in input e che aggiorna il contatore degli eventi in memoria. A tal proposito si assuma che i registri da R20 a R30 possano essere utilizzati senza la necessità di dover preservare il loro contenuto

Esercizio 2

Indicare come potrebbe essere la struttura *ai morsetti* di un register file del DLX evidenziando e descrivendo sinteticamente il significato di tutti i segnali.

Esercizio 3

Come noto il DLX non emette i segnali d'indirizzo BA1 e BA0. Se non fosse così, sarebbe possibile generare i segnali BE3, BE2, BE1 e BE0 utilizzando esclusivamente i segnali BA1 e BA0? Esplicitare chiaramente le motivazioni della risposta.

MAPPING

EPROM 1 KB : $0 \times 0000\ 0000 \rightarrow 0 \times 000\text{FFFF}$ (4×256 KB)

RAM 2 KB : $0 \times \text{FFD}00000 \rightarrow 0 \times \text{FFFC}\text{FFFF}$ (4×512 KB)

CS-EN-COUNT : $0 \times \text{FFFFFFF0}$

CS-EPROM₋₀ = $\overline{\text{BA31 BA30}}$ BE0

CS-EPROM₋₁ = $\overline{\text{BA31 BA30}}$ BE1

CS-EPROM₋₂ = $\overline{\text{BA31 BA30}}$ BE2

CS-EPROM₋₃ = $\overline{\text{BA31 BA30}}$ BE3

CS-RAM₋₀ = BA31 BE0

CS-RAM₋₁ = BA31 BE1

CS-RAM₋₂ = BA31 BE2

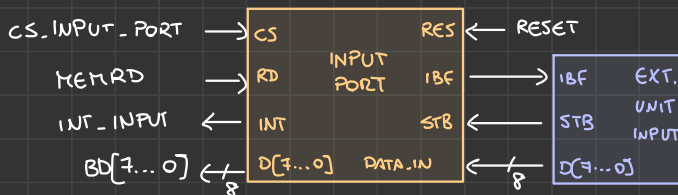
CS-RAM₋₃ = BA31 BE3

CS-INPUT-PORT = $\overline{\text{BA31 BA30}}$ BE0

CS-READ-SUM = BA31 BA30 BE1

CS-READ-INT = $\overline{\text{BA31}}$ BA30 BE2

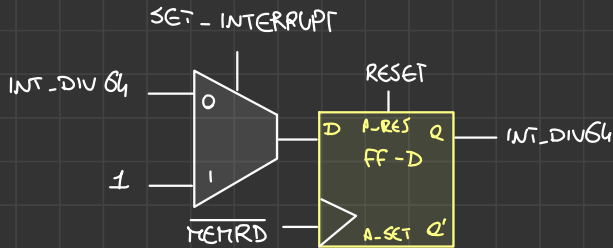
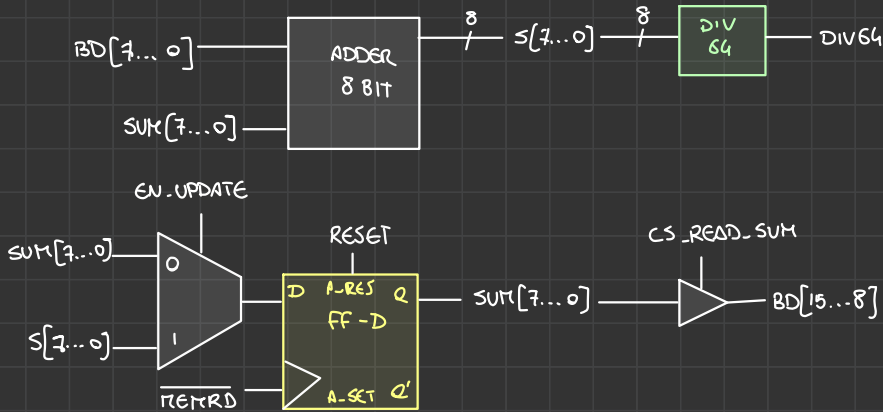
PORTA INPUT



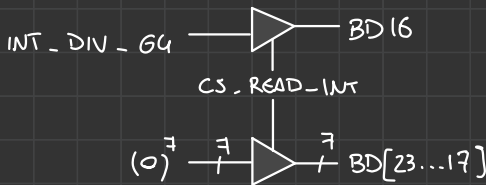
RETE PER LA SOMMA MODULO 256



Rete per la somma



$$SET_INTERRUPT = ENABLE_UPDATE \cdot READ_7 \cdot DIV64$$



$$INT(TO\ DLX) = INT_INPUT_PORT + INT_DIV_64$$

CODICE

```
0h    LHI    R20, 0x4000
4h    LBU    R21, 0x0002(R20)
8h    BNEZ   R21, DIV_64
Ch    LBU    R22, 0x0000(R20)    DIV_64 = 8h
10h    RFE
14h    DIV_64 LHI    R25, 0xFFFF
18h                LW    R26, 0xFFFF0(R25)
1Ch                ADDI   R27, R26, 0x0001
20h                SW     R27, 0xFFFF0(R25)

24h                LBU    R30, 0x0001(R20)
28h                SB     R30, 0xFFFF4(R25)
2Ch                RFE;
```